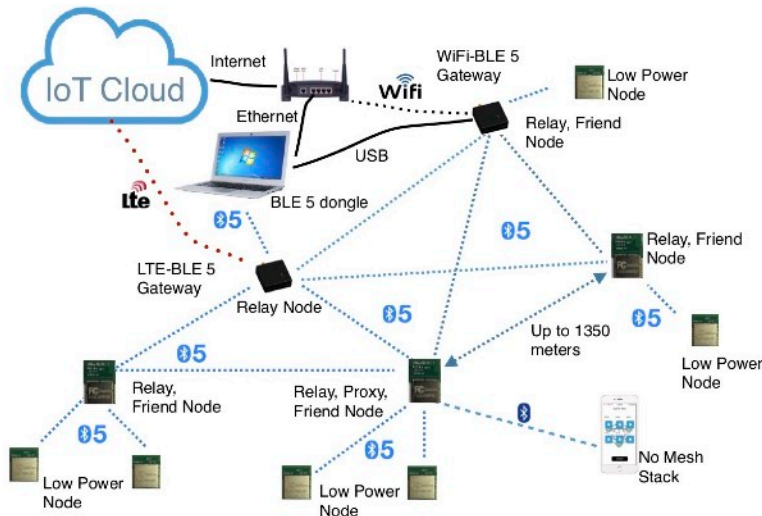


### OpenSource with Market Ready Enclosure

Bluetooth 5 to WiFi Gateway (BWG) provides connection from Bluetooth devices to cloud server. They are in market ready enclosure and pre-certified. OpenSource codes are available to expedite your application firmware and software development.



### Gateway for Home/Small Mesh

BWG832F is a low cost Bluetooth 5-WiFi gateway for point to point or small mesh network where security is not required or limited decryption and authentication can be implemented in firmware. Its range to other node is 280 meters, enough for home and typical office implementation.

### BWG832X to Penetrate Multiple Walls

Using BT832X with +20 dBm TX, BWG832X can penetrate multiple walls or be used in wide area Bluetooth mesh applications. With an external antenna,

### Nordic nRF52 Development Environment

Nordic nRF52 SoC in Bluetooth module communicates with a cloud server in message layers. Nordic nRF52 development environment should be used. Source codes

specific to gateway interfacing with Fanstel BLE 5 sensors are available at:

<http://www.fanstel.com/download-opensource/>

### Over-The-Air DFU

The Bluetooth module is supported by an Over-The-Air Device Firmware Upgrade (OTA DFU) feature. This allows for in the field updates of application software and SoftDevice.

### WiFi Module Development-Mongoose OS

**Development Kit (e.g., DK-BWG832F) is required to load codes into WiFi module.** DK is preloaded with codes for connection to Fanstel development server.

**Mongoose OS** OpenSource codes for connecting to Fanstel development server and to **Google Cloud IoT Core** are available. They can be used as base to develop interface with other servers.

Link to download additional document and source codes:

<http://www.fanstel.com/download-opensource/>

### Miscellaneous

- Size: 60x60x22mm
- Desk and wall mountable
- Power supply and USB cable included
- 2 LED indicators defined by user
- Operation temperature: -40°C to +85°C
- FCC ID: X8WBT832
- IC ID: 4100A-BT832
- CE:
- QDID: 97989



### Gateway Summaries

Gateway	BWG832F	BWG832X
WiFi module	ESP 8266	ESP 8266
BLE module	BT832F	BT832X
Flash/RAM	512KB/64KB	512KB/64KB
BT range 1 Mbps	280 meters	1170 meters
Development Kit	DK-BWG832F	DK-BWG832X

## Table Of Contents

1. Introduction.....	3
BWG832F .....	3
BWG832X.....	3
BWG832XE .....	3
Hardware and Accessories Required for Firmware Development.....	3
2. Gateway and Development Kit.....	4
Development Kit (DK).....	4
Gateway Development Kit Block Diagram.....	5
Gateway .....	6
Gateway Block Diagram .....	6
Testing Connection to Fanstel Mosquito Server .....	7
3. Develop Codes for WiFi Module ESP8266 .....	10
Fanstel Mosquito Development Server .....	10
Google Cloud IoT Core.....	10
Preparing Gateway Board for Programming.....	11
Loading Firmware into WiFi Module.....	12
Connecting to Fanstel Mosquito Server .....	14
Connecting to Google Cloud Platform IoT Core.....	15
Programming WiFi Module in Production Gateway.....	17
4. Develop Codes for Bluetooth Modules.....	18
Nordic Development Environment.....	18
Interfacing with Nordic Thingy .....	18
Fanstel Sensors.....	19
Interfacing with a Fanstel Evaluation Board .....	19
Check COM Port Connection .....	20
Starting Nordic nRFgo .....	20
Over-The-Air DFU .....	21
Control Skyworks Power Amplifier.....	21
Programming Bluetooth Module in Production Gateway.....	24
5. BWG832F/X Gateway Schematics .....	25
6. Miscellaneous.....	26
FCC Label .....	26
Revision History .....	27
Contact Us .....	28

## 1. Introduction

Bluetooth 5 to WiFi Gateway (BWG) provides connection from Bluetooth devices to cloud server. They are in market ready enclosure and pre-certified. OpenSource codes are available to expedite your application firmware and software development.

An ESP8266 WiFi module provide connection to cloud server. A Nordic nRF52832 module work as a Relay/Proxy node in a Bluetooth mesh, relaying message between a cloud server and other nodes in mesh network.

### **BWG832F**

- ESP8266 WiFi module
- BT832F with Nordic nRF52832 SoC
- Bluetooth range between nodes: 280 meters

### **BWG832X**

- ESP 8266 WiFi module
- BT832X with Nordic nRF52832 SoC and SKY66112 power amplifier
- Bluetooth range between nodes: 1170 meters

### **BWG832XE**

- ESP 8266 WiFi module
- BT832X with Nordic nRF52832 SoC, SKY66112 power amplifier, external antenna
- Bluetooth range between nodes: 1170 meters BWG832F Block Diagram

## **Hardware and Accessories Required for Firmware Development**

The following are required to develop firmware for gateway WiFi and Bluetooth modules.

- Development Kit version of gateway. It contains accessories required to load WiFi firmware.
- A Nordic nRF52DK
- A TAG Connect cable to load firmware into gateway Bluetooth module.

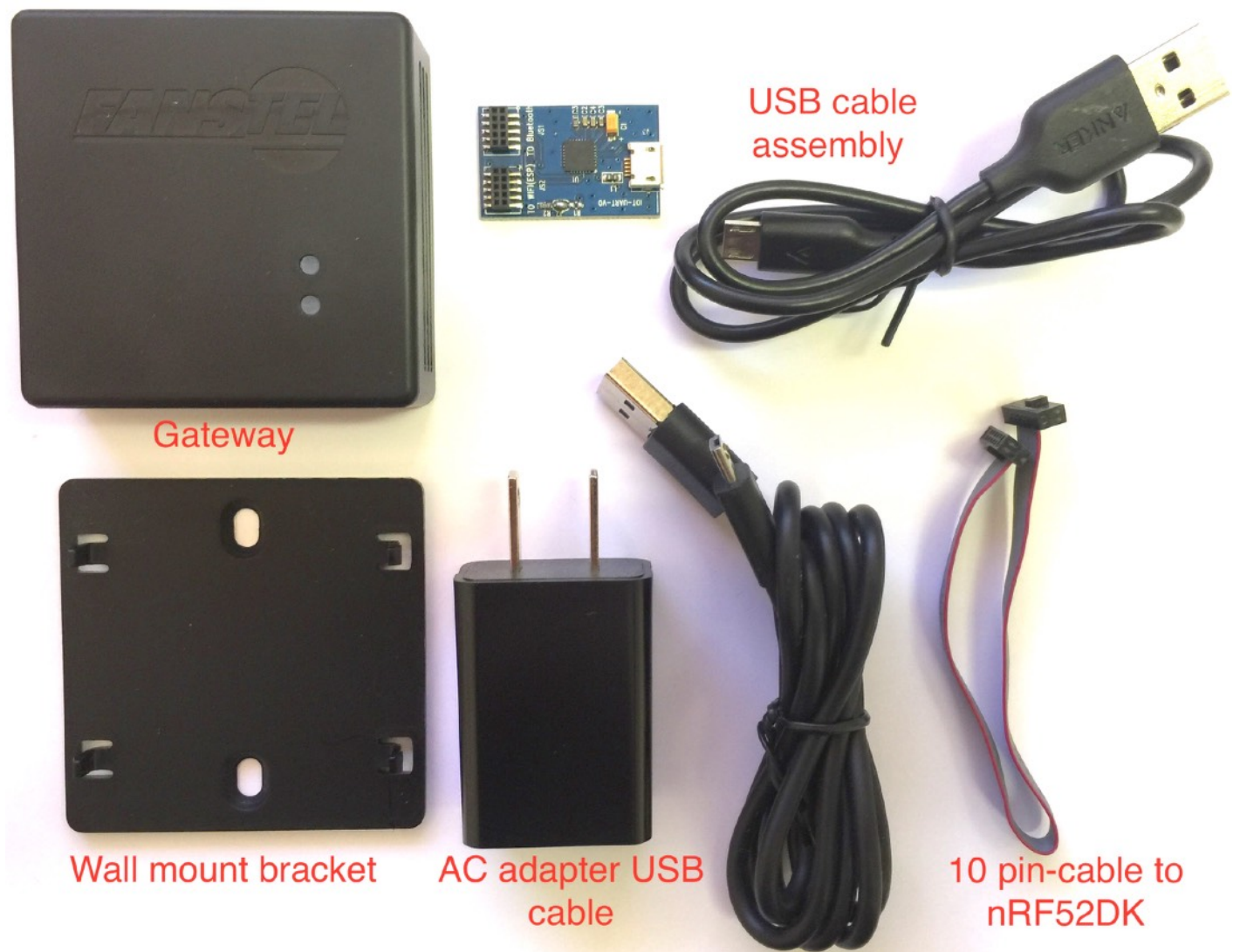
# BWG832F/X/XE Bluetooth 5 to WiFi Gateway

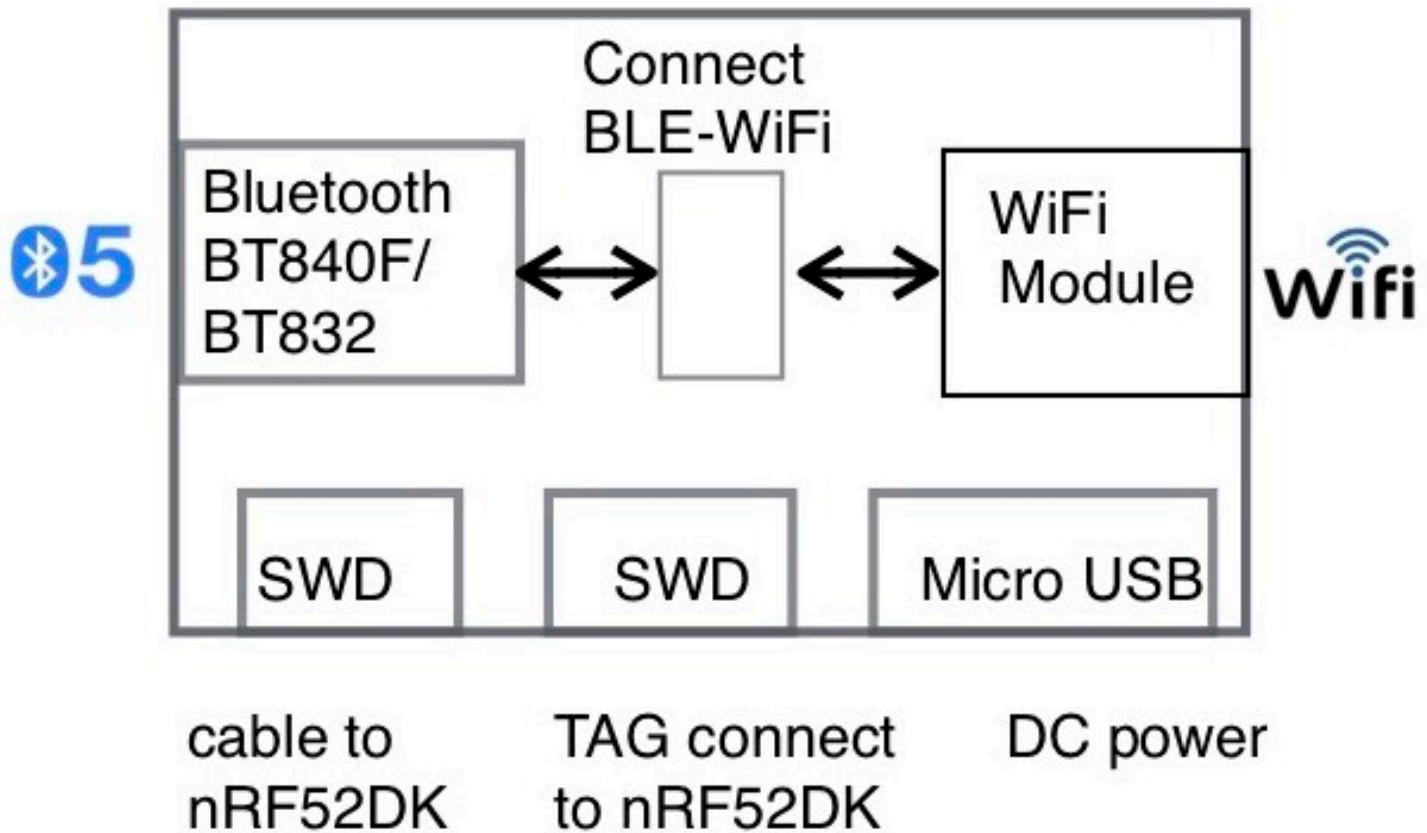
## 2. Gateway and Development Kit

### Development Kit (DK)

A Development Kit, for example, DK-BWG832F is required to load firmware into WiFi module. A DK consists of the followings:

- A gateway with optional magnet
- A wall mount bracket. Don't install wall mount bracket for magnetic mounting
- An USB cable assemble including a cable and an USB to UART bridge board.
- An AC adapter with USB cable
- 10-pin flat cable for connection to Nordic nRF52DK (required, not included)



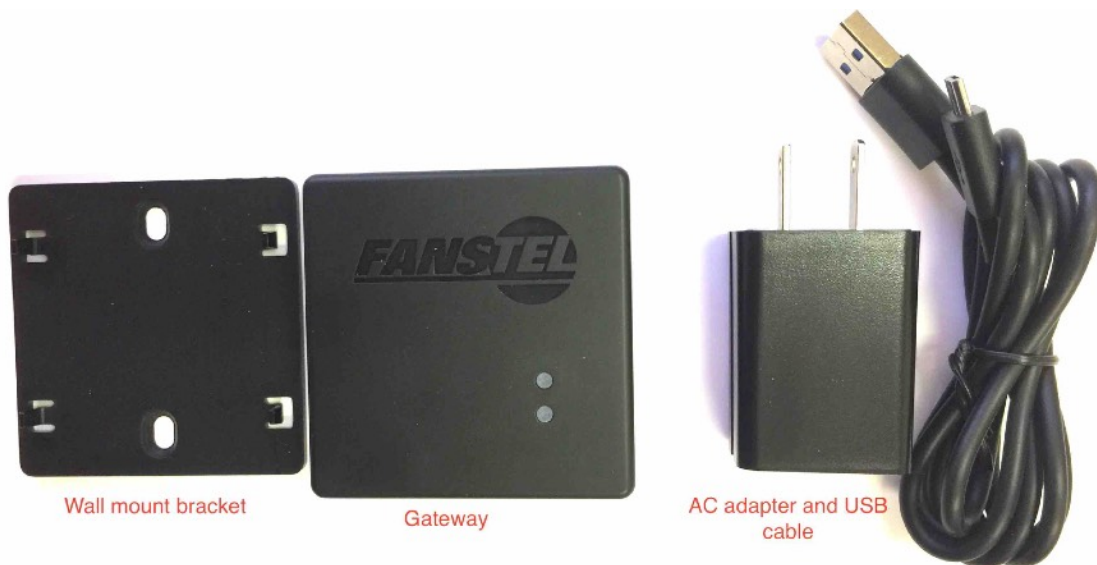
**Gateway Development Kit Block Diagram**

A gateway development kit has the following main components:

- A Fanstel Bluetooth 5 module
- An ESP8266 WiFi module
- Connect BLE-WiFi block allowing connection between (a) a PC and WiFi module (b) a PC and Bluetooth module (c) WiFi module and Bluetooth module.
- A micro USB connector for DC power input
- A 10-pin flat cable connector to connect to a Nordic nRF52DK for loading firmware into Bluetooth 5 module.
- A 10-pin pad for TAG connect cable TC2050-IDC-NL-050-ALL to connect to a Nordic nRF52DK for loading firmware into Bluetooth 5 module.

## Gateway

A gateway package consists of the followings:

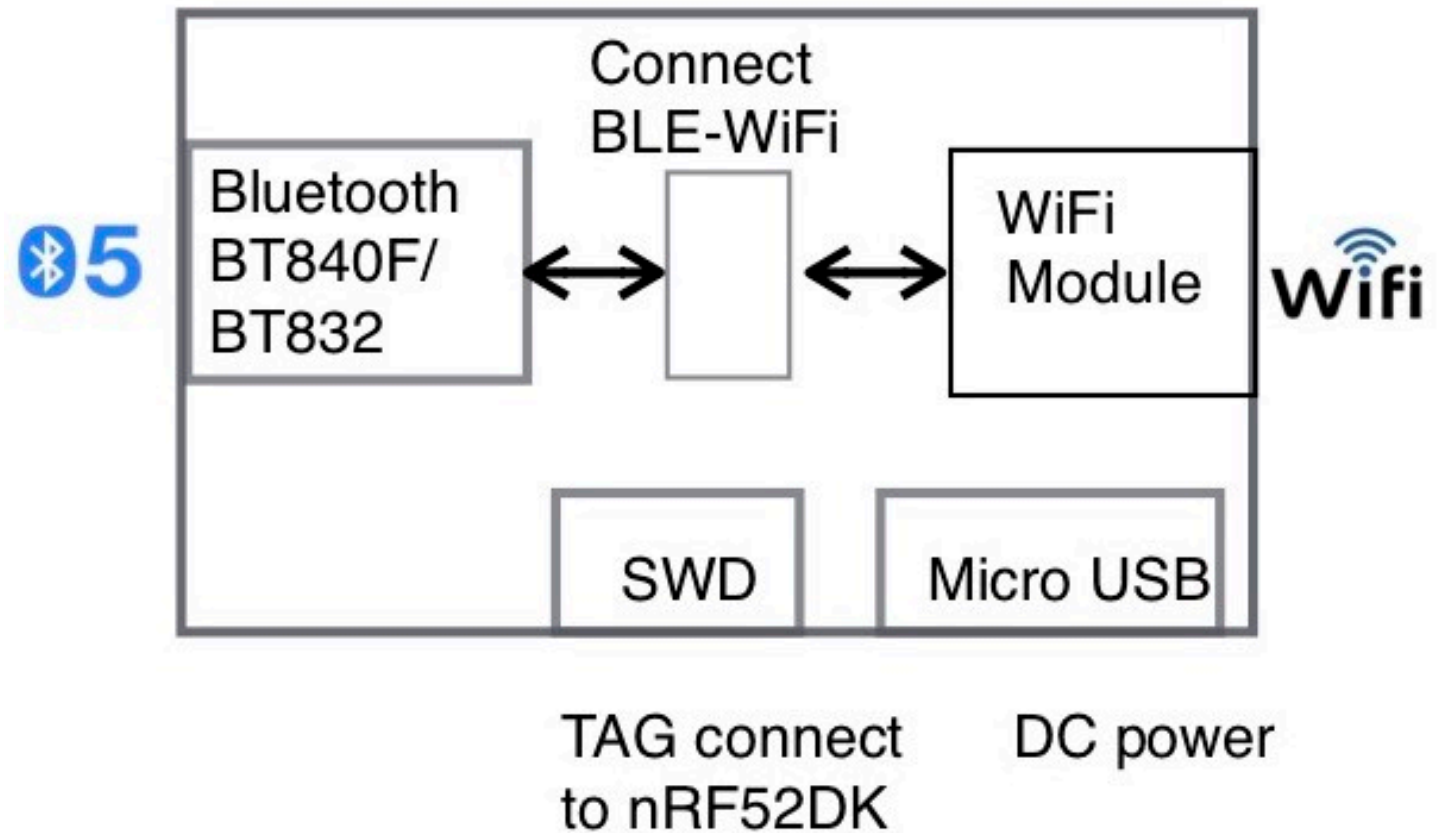


- A gateway
- A wall mount bracket.
- An AC adapter with USB cable

## Gateway Block Diagram

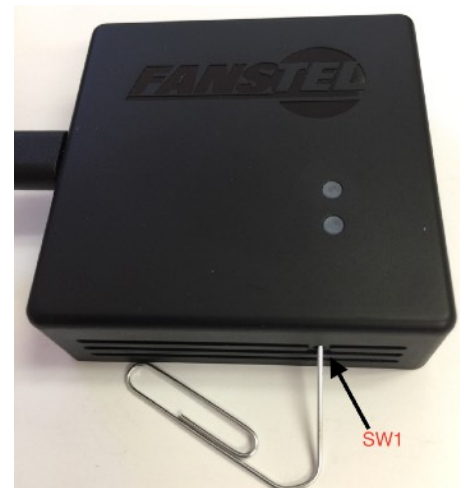
A gateway has the following main components:

- A Fanstel Bluetooth 5 module
- An ESP8266 WiFi module
- Connect BLE-WiFi block allowing connection between (a) a PC and WiFi module (b) a PC and Bluetooth module (c) WiFi module and Bluetooth module. (The USB cable assemble included in DK is required for PC connection).
- A micro USB connector for DC power input
- A 10-pin pad for TAG connect cable TC2050-IDC-NL-050-ALL to connect to a Nordic nRF52DK for loading firmware into Bluetooth 5 module.



### Testing Connection to Fanstel Mosquito Server

1. Set WiFi module to Access Point (AP) mode. .
  - Use tip of a paper clip to press and hold SW1 when connecting USB power supply to boot up.
2. Use a smartphone to connect to WiFi AP  
FanstelGW\_XXXXXX  
The XXXXXX is the device ID, 7A561D in the example.  
Please remember this ID.





3. Open the web browser then go to 192.168.4.1  
Enter your router information at the web page.
4. The Nordic and ESP chip's LED will flash when power up.  
The baudrate to 115200 to check the log.  
You should the detail from log.

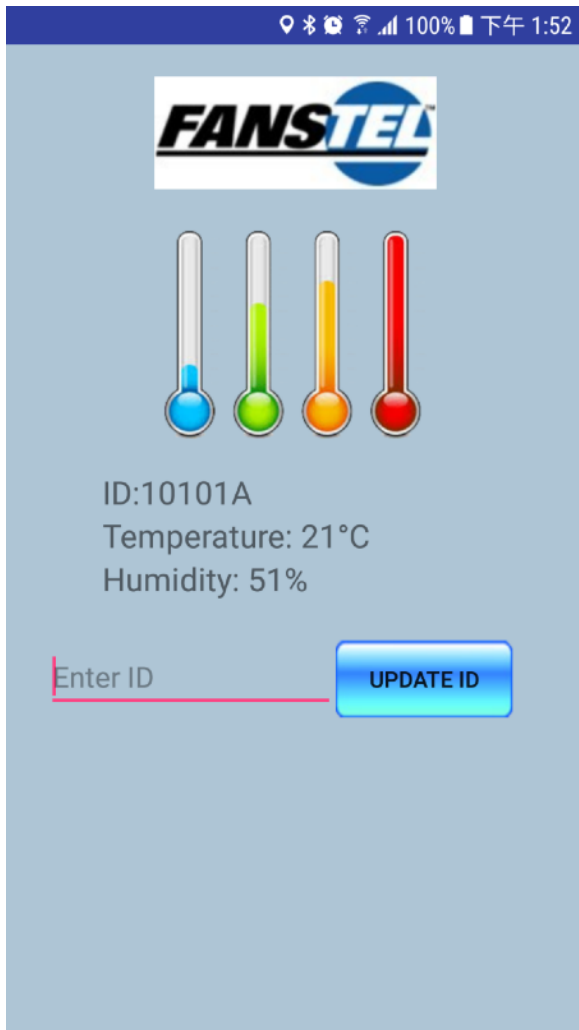
```

esp_mgos_init2      Mqtt_Demo 1.0 (20180321-071019/???)<LF>
esp_mgos_init2      Mongoose OS 2018032107 (20180321-071018/???)<LF>
esp_mgos_init2      CPU: ESP8266EX, 160 MHz, RAM: 52000 total, 46896 free<LF>
esp_mgos_init2      SDK 2.2.0-dev(e6d3ebb); flash: 4M<LF>
esp_print_reset_info Reset cause: 0 (power on)<LF>
mgos_vfs_dev_open   sysflash () -> 0x3fff08f4<LF>
mgos_vfs_mount      Mount SPIFFS @ / (dev 0x3fff08f4, opts {"addr": 32768, "size": 262144}
<LF>
mgos_vfs_mount      /: size 233681, used: 149345, free: 84336<LF>
mgos_sys_config_init MAC: 2E3AE818173F<LF>
mgos_sys_config_init WDT: 30 seconds<LF>
mgos_mdns_init      Listening on udp://:5353<LF>
mgos_mdns_hal_join_g Joining multicast group 224.0.0.251<LF>
bcn 0<LF>
del if1<LF>
del if0<LF>
usl<LF>
mode : null<LF>
mgos_wifi_setup     WiFi mode: off<LF>
mgos_http_server_ini HTTP server started on [80]<LF>
mgos_ota_shadow_init OTA over shadow initialised<LF>
mgos_ota_http_client Init done, ota_lib=(null), ota handler 1<LF>
mg_rpc_channel_mqtt 0x3fff022c esp8266_18173F/rpc/#<LF>
mg_rpc_channel_uart 0x3fff1524 UART0<LF>
mgos_init           Init done, RAM: 52000 total, 42156 free, 40016 min free<LF>
mgos_aws_shadow_init MQTT is not configured for AWS, not initialising shadow<LF>
mongoose_poll       New heap free LWM: 28184<LF>

```

5. Install the Android APP and input the ID.
6. Now the connection is link between the gateway and phone.





## 3. Develop Codes for WiFi Module ESP8266

**Your first development gateway must be a Development Kit (DK).** DK includes all accessories for WiFi interface development. For Bluetooth interface development, a Nordic nRF52DK is required, in addition.

All DK gateways are preloaded with firmware for connecting to a Fanstel server. Please follow procedures in the previous section to test connection to the Fanstel server before developing and loading your codes into gateway.

WiFi module is used to set up and maintain connection to a cloud server through a WiFi router.

**Mongoose OS** OpenSource codes for connecting to **Fanstel Mosquitto development server** and to **Google Cloud Platform IoT Core** are available. They can be used as base to develop interface with other servers. Link to download additional document, hex, and source codes:

<http://www.fanstel.com/download-open-source/>

**MQTT** (MQ Telemetry Transport), was also known as Message Queuing Telemetry Transport) protocols are used to interface with cloud server.

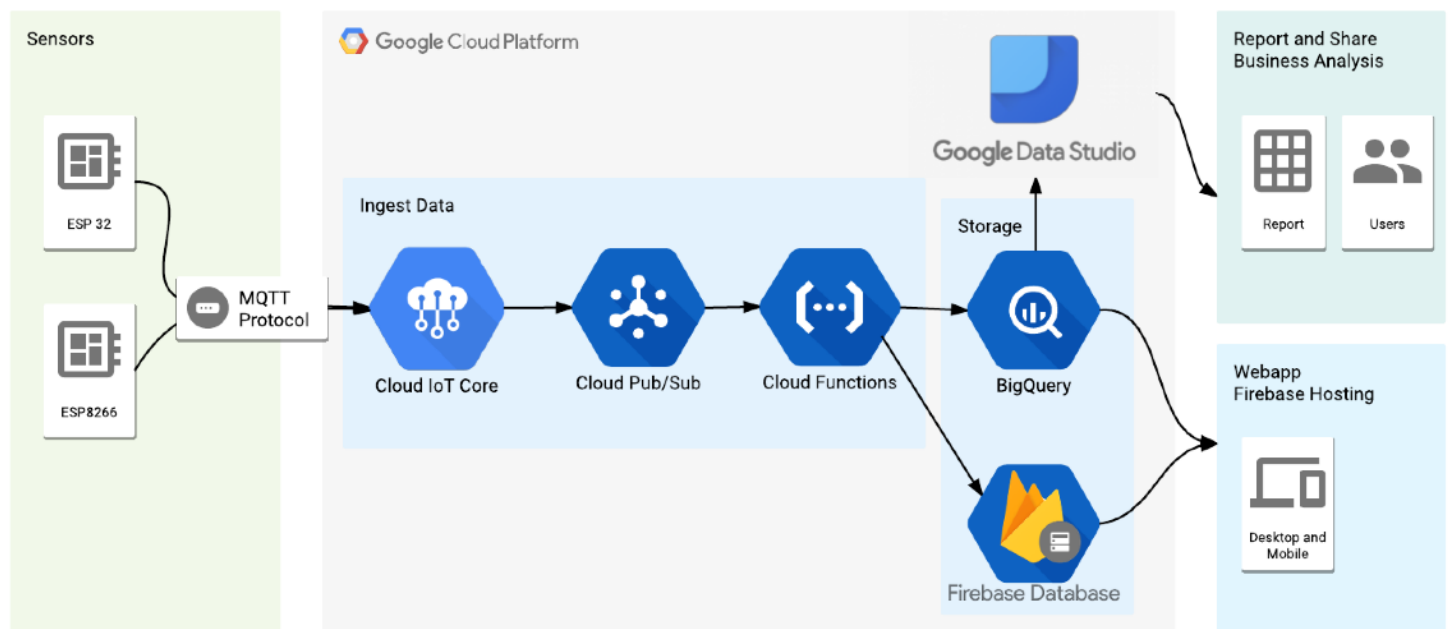
### Fanstel Mosquitto Development Server

WiFi modules in DK is preloaded with firmware connecting to Fanstel development server. Fanstel team can better assist developers on trouble-shooting interface issues in the early days of development.

Upon successfully test your codes with Fanstel development server, please use other servers instead.

### Google Cloud IoT Core

A block diagram for interfacing ESP 8266 with Google Cloud IoT Core is shown.



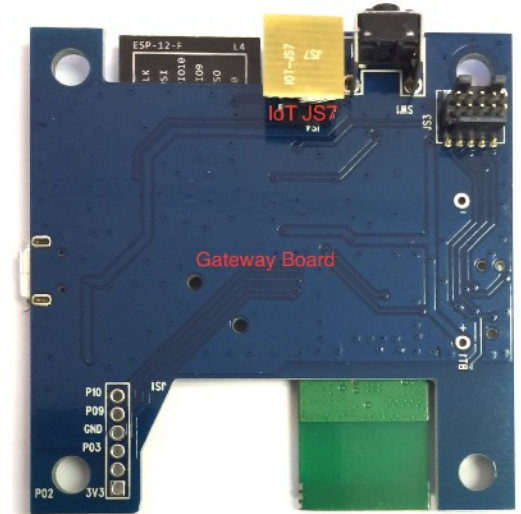
# BWG832F/X/XE Bluetooth 5 to WiFi Gateway

Draft Ver 0.60 July 2018

## Preparing Gateway Board for Programming

A. Remove 4 screws from bottom side of enclosure and remove gateway board from enclosure.

The small PCBA IoT-JS7 is for connecting WiFi module UART port to Bluetooth module UART port. Remove it from gateway board..

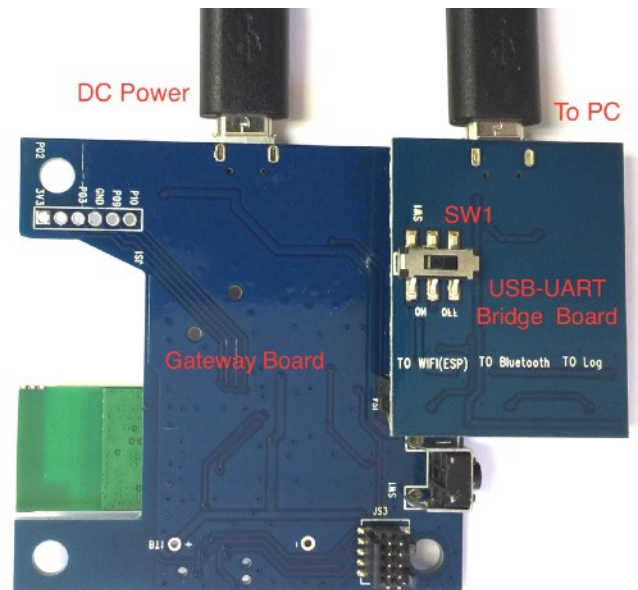


B. USB to UART bridge board has 3 female connectors. Connect to

- TO WiFi (ESP), JS2: for connecting WiFi module to a PC USB port
- TO Bluetooth, JS1: for connecting Bluetooth module to a PC USB port.
- TO log, JS3.

Connect the USB-UART Bridge Board JS2 to Gateway Board JS4. Switch SW1 can be used to set boot mode of WiFi module.

- Set SW1 to **ON** for **UART-Boot mode**, or loading firmware from a PC to WiFi module during booting up.
- Set SW1 to **OFF** for **Flash-Boot mode**, or booting up from flash memory in normal operation.



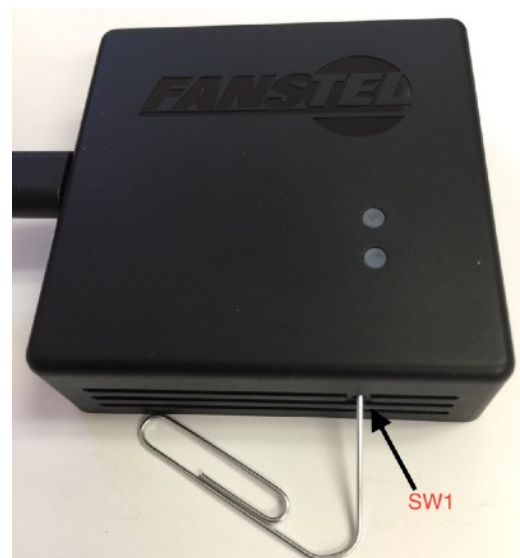
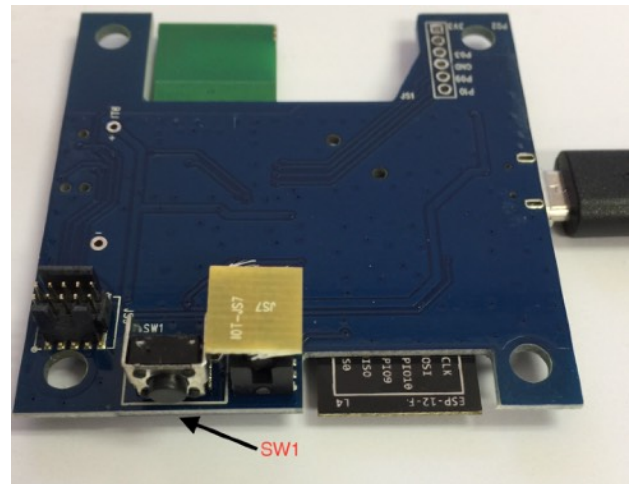
## Loading Firmware into WiFi Module

7. Copy the mos.exe and build in the same folder.
8. Connect USB-UART Bridge Board to a PC with an USB cable.
9. In the Window command mode  
Type command: `mos flash --port COM5`  
The COM port number can be different in your computer.
10. If everything is good.  
It will show “All done!”.

```
Running flasher @ 460800...
Flasher is running
Flash size: 4194304, params: 0x0240 (dio,32m,40m)
Deduping...
 2736 @ 0x0 -> 0
262144 @ 0x8000 -> 28672
778528 @ 0x100000 -> 0
  128 @ 0x3fc000 -> 0
Writing...
 4096 @ 0x7000
 8192 @ 0x1e000
20480 @ 0x2c000
 4096 @ 0x3fb000
Wrote 36864 bytes in 0.44 seconds (657.53 KBit/sec)
Verifying...
 2736 @ 0x0
 4096 @ 0x7000
262144 @ 0x8000
778528 @ 0x100000
 4096 @ 0x3fb000
  128 @ 0x3fc000
Booting firmware...
All done!
```

## 11. Remove the USB-UART Bridge Board from Gateway Board.

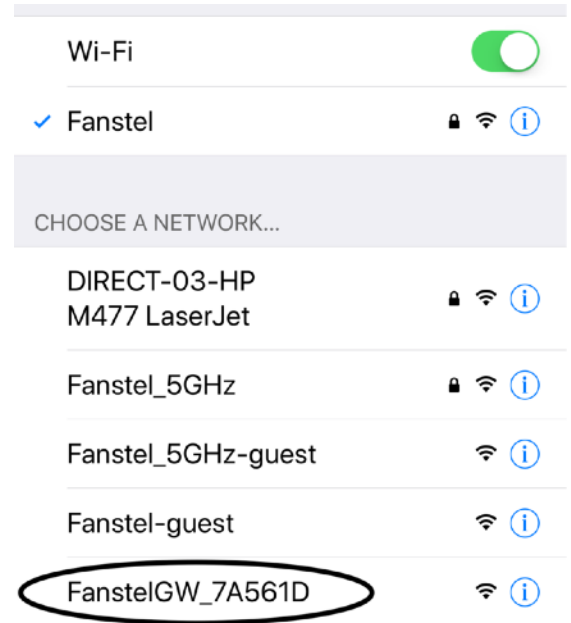
- Re-install the small IoT-JS7 board to JS4 of Gateway board. IoT -JS7 board is symmetrical. It can be installed in either direction.
- Hold the Gateway Board SW1 button when connecting USB power supply to boot up.
- If Gateway Board is inside enclosure, use tip of a paper clip to press and hold SW1 when connecting USB power supply to boot up.



12. Use a smartphone to connect to WiFi AP

FanstelGW\_XXXXXX

The XXXXXX is the device ID, 7A561D in the example.  
Please remember this ID.



13. Open the web browser then go to 192.168.4.1

Enter your router information at the web page.

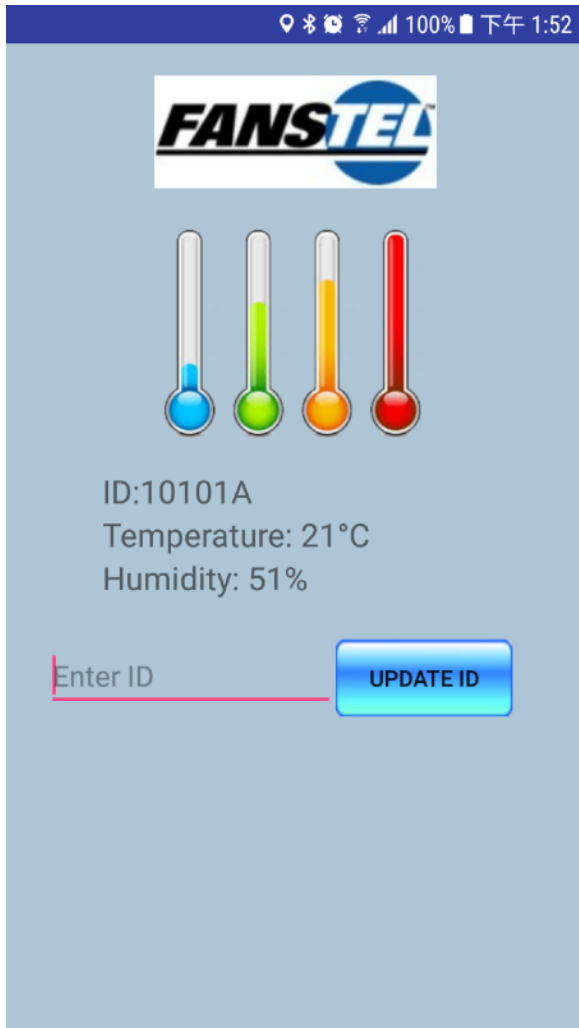
14. The Nordic and ESP chip's LED will flash when power up.

The baudrate to 115200 to check the log.

You should the detail from log.

```
esp_mgos_init2      Mqtt_Demo 1.0 (20180321-071019/???)<LF>
esp_mgos_init2      Mongoose OS 2018032107 (20180321-071018/???)<LF>
esp_mgos_init2      CPU: ESP8266EX, 160 MHz, RAM: 52000 total, 46896 free<LF>
esp_mgos_init2      SDK 2.2.0-dev(e6d3ebb); flash: 4M<LF>
esp_print_reset_info Reset cause: 0 (power on)<LF>
mgos_vfs_dev_open   sysflash () -> 0x3fff08f4<LF>
mgos_vfs_mount      Mount SPIFFS @ / (dev 0x3fff08f4, opts {"addr": 32768, "size": 262144}
<LF>
mgos_vfs_mount      /: size 233681, used: 149345, free: 84336<LF>
mgos_sys_config_init MAC: 2E3AE818173F<LF>
mgos_sys_config_init WDT: 30 seconds<LF>
mgos_mdns_init      Listening on udp://:5353<LF>
mgos_mdns_hal_join_g Joining multicast group 224.0.0.251<LF>
bcn 0<LF>
del if1<LF>
del if0<LF>
us1<LF>
mode : null<LF>
mgos_wifi_setup     WiFi mode: off<LF>
mgos_http_server_ini HTTP server started on [80]<LF>
mgos_ota_shadow_init OTA over shadow initialised<LF>
mgos_ota_http_client Init done, ota_lib=(null), ota handler 1<LF>
mg_rpc_channel_mqtt 0x3fff022c esp8266_18173F/rpc/#<LF>
mg_rpc_channel_uart 0x3fff1524 UART0<LF>
mgos_init           Init done, RAM: 52000 total, 42156 free, 40016 min free<LF>
mgos_aws_shadow_init MQTT is not configured for AWS, not initialising shadow<LF>
mongoose_poll       New heap free LWM: 28184<LF>
```

15. Install the Android APP and input the ID.
16. Now the connection is link between the gateway and phone.



## Connecting to Fanstel Mosquitto Server

Fanstel server is a Mosquitto free MQTT server.

**MOS\_MQTT\_Auto** publishes messages to MQTT broker in Fanstel server. These messages contains preset (artificial) sensor data. A Bluetooth sensor is not required to generate these sensor data.

**MOS\_MQTT\_SensorTDK** publishes messages to MQTT broker in Fanstel server. These messages contains real sensor data. A Fanstel or other BLE sensor box is required.

The Android APP subscribes to the same topic. Sensor data can be displayed on an Android device.

You might want to build your own MQTT server. The followings are basic Mosquitto set up information.

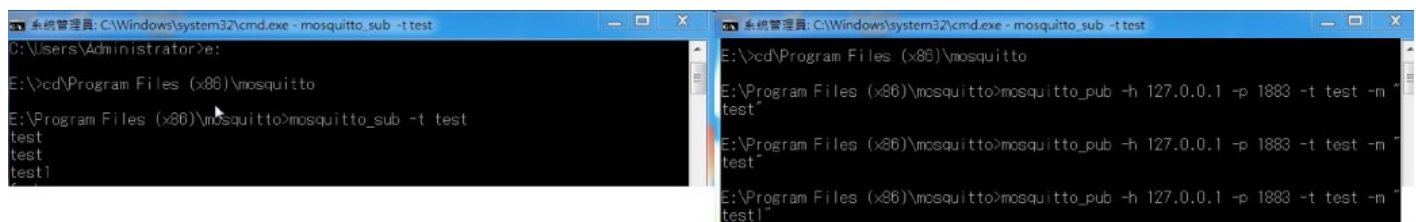
To install:

1. Download Win32OpenSSL V1.02  
<http://siproweb.com/products/Win32OpenSSL.html>  
Select bin when install. After install, copy libeay32.dll and ssleay32.dll to mosquito install folder.
2. Download pthreadVC2.dll and copy the file to mosquito install folder.  
<ftp://sources.redhat.com/pub/pthreads-win32/dll-latest/dll/x86/>
3. Download mosquito native build version  
mosquito-1.4.14-install-win32.exe (~200 kB) (Native build, Windows Vista and up, built with Visual Studio Community 2015)

Running mosquito.exe. The server started run as broker.

Please also make sure your fire wall is setup to pass the port 1883.

Open two DOS windows, one runs as Subscriber another one runs as Publisher



```
System Administrator: C:\Windows\system32\cmd.exe - mosquito_sub -t test
D:\Users\Administrator>e:
E:\>cd\Program Files (x86)\mosquito
E:\Program Files (x86)\mosquito>mosquito_sub -t test
test
test
test1

System Administrator: C:\Windows\system32\cmd.exe - mosquito_sub -t test
E:\>cd\Program Files (x86)\mosquito
E:\Program Files (x86)\mosquito>mosquito_pub -h 127.0.0.1 -p 1883 -t test -m "test"
E:\Program Files (x86)\mosquito>mosquito_pub -h 127.0.0.1 -p 1883 -t test -m "test"
E:\Program Files (x86)\mosquito>mosquito_pub -h 127.0.0.1 -p 1883 -t test -m "test!"
```

Publisher: `mosquito_pub -h YOUR ADDRESS -p 1883 -t YOUR TOPIC -m "YOUR MESSAGE"`

Subscriber: `mosquito_sub -h YOUR ADDRESS -t YOUR TOPIC`

## Connecting to Google Cloud Platform IoT Core

Use the following procedures to connect gateway to Google Cloud Platform (GCP).

### GCP Server setting:

Install the gcloud.

# Install beta components:

`gcloud components install beta`

# Authenticate with Google Cloud:

`gcloud auth login`

# Create cloud project – choose your unique project name:

```
gcloud projects create YOUR_PROJECT_NAME
```

```
# Add permissions for IoT Core
```

```
gcloud projects add-iam-policy-binding YOUR_PROJECT_NAME --member=serviceAccount:cloud-  
iot@system.gserviceaccount.com --role=roles/pubsub.publisher
```

```
# Create PubSub topic for device data:
```

```
gcloud beta pubsub topics create telemetry-topic
```

```
# Create PubSub subscription for device data:
```

```
gcloud beta pubsub subscriptions create --topic telemetry-topic telemetry-subscription
```

```
# Create device registry:
```

```
gcloud beta iot registries create weather-station-registry --region us-central1 --event-pubsub-  
topic=telemetry-topic
```

### **Gateway side setting:**

```
# Get project ID of your new project:
```

```
gcloud projects list
```

```
mos gcp-iot-setup --gcp-project YOUR_PROJECT_NAME --gcp-region us-central1 --gcp-registry  
YOUR_REGISTRY
```

```
mos gcp-iot-setup --gcp-project YOUR_PROJECT_ID --gcp-region europe-west1 --gcp-registry iot-  
registry
```

```
mos gcp-iot-setup --gcp-project fansteliot180409 --gcp-region us-central1 --gcp-registry weather-  
station-registry
```

### **Android APP setting:**

```
#APP build.gradle
```

```
// ADD THIS AT THE BOTTOM
```

```
apply plugin: 'com.google.gms.google-services'
```

```
dependencies {
```

```
.....
```

```
compile 'com.google.firebase:firebase-core:12.0.1'
```

```
compile 'com.google.firebase:firebase-database:12.0.1'
```

```
}
```

```
#Project build.gradle
```

```
dependencies {
```

```
.....
```

```
classpath 'com.google.gms:google-services:3.2.0'
```

```
}
```

```
allprojects {
```

```
repositories {
```



```
// ...
maven {
    url "https://maven.google.com" // Google's Maven repository
}
}
```

#download google.json file copy to app/ folder  
need to setup the project path.

#### **MOS basic command**

mos flash -port COM5 =>programming the firmware  
mos -port COM5 =>Open the GUI via Serial Port  
mos -port ws://192.168.0.188/rpc =>Open the GUI via web  
mos wifi SSID PWD =>Setup WiFi  
mos config-get => Get on board config file.

### **Programming WiFi Module in Production Gateway**

The USB cable assembly included in the Development Kit can be used to load firmware into WiFi module of production gateway.

For high volume production, WiFi firmware can be preloaded in factory.

## 4. Develop Codes for Bluetooth Modules

Gateway has a Nordic nRF52832 module.

- BWG832F with BT832F, Nordic nRF52832 with an integrated antenna.
- BWG832X with BT832X, Nordic nRF52832, SKYWOKS SKY66112, and an integrated antenna.
- BWG832XE with BT832XE, Nordic nRF52832, SKYWOKS SKY66112, and an u.FL connector for an external antenna.

### Nordic Development Environment

Nordic Semiconductor provides a complete range of hardware and software development tools for the nRF52 Series devices. nRF52 DK board is recommended for firmware development. Document and Software development tools can be downloaded by the following links.

#### Get start with Nordic chip and all online documents.

[http://infocenter.nordicsemi.com/index.jsp?topic=/com.nordic.infocenter.nrf52/dita/nrf52/development/nrf52\\_dev\\_kit.html&cp=1\\_1](http://infocenter.nordicsemi.com/index.jsp?topic=/com.nordic.infocenter.nrf52/dita/nrf52/development/nrf52_dev_kit.html&cp=1_1)

#### Nordic SDK with many example projects.

[https://developer.nordicsemi.com/nRF5\\_SDK/](https://developer.nordicsemi.com/nRF5_SDK/)

#### Nordic development zone. You can search or ask a question there.

<https://devzone.nordicsemi.com/tutorials/b/getting-started/posts/development-with-gcc-and-eclipse>

#### Programming the Nordic chip

Download and install Nrf5x-Command-Line Tools

<https://www.nordicsemi.com/eng/nordic/Products/nRF52840/nRF5x-Command-Line-Tools-Win32/58850>

#### Download and install nRFgo Studio

<https://www.nordicsemi.com/eng/Products/2.4GHz-RF/nRFgo-Studio/>

### Interfacing with Nordic Thingy

During gateway BLE firmware development, Nordic Thingy can be used as sources of sensor data. Interfacing with

## BWG832F/X/XE Bluetooth 5 to WiFi Gateway

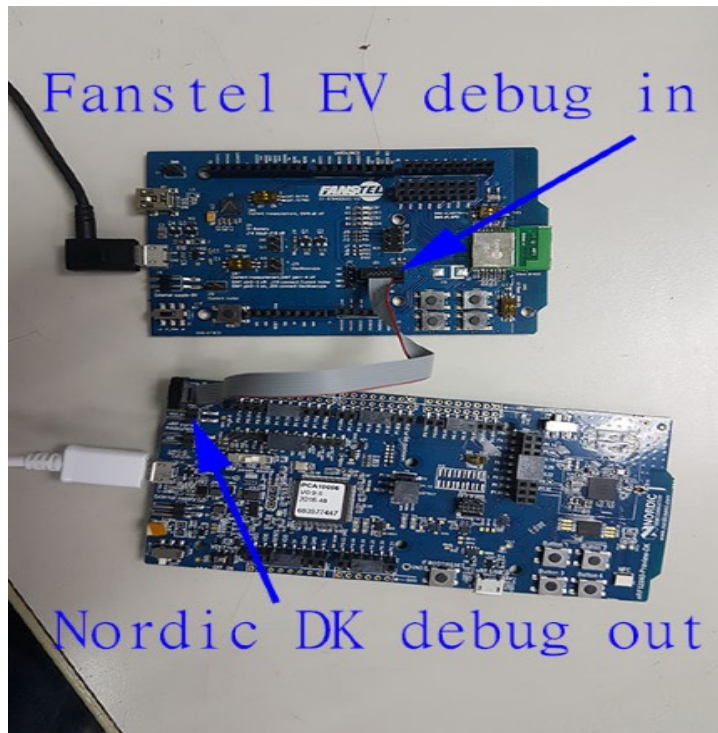
### Fanstel Sensors

Fanstel sensors are under development.

### Interfacing with a Fanstel Evaluation Board

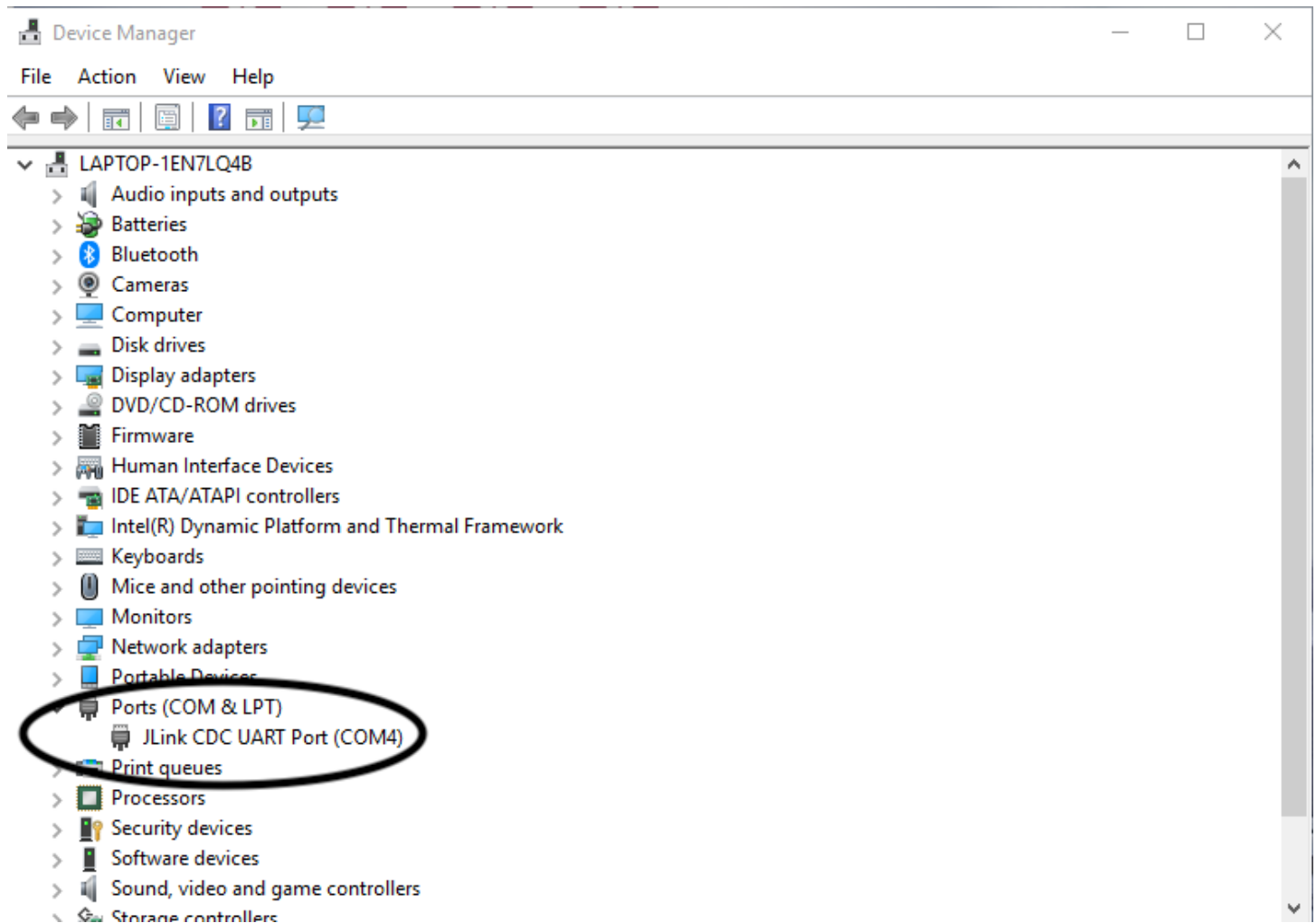
If you have a Fanstel nRF52 module evaluation board.

- Connect Nordic nRF52DK debug out to Fanstel EV debug in as below.
- Connect Nordic DK to PC.
- Connect a DC power source to micro or mini USB port of Fanstel board.



## Check COM Port Connection

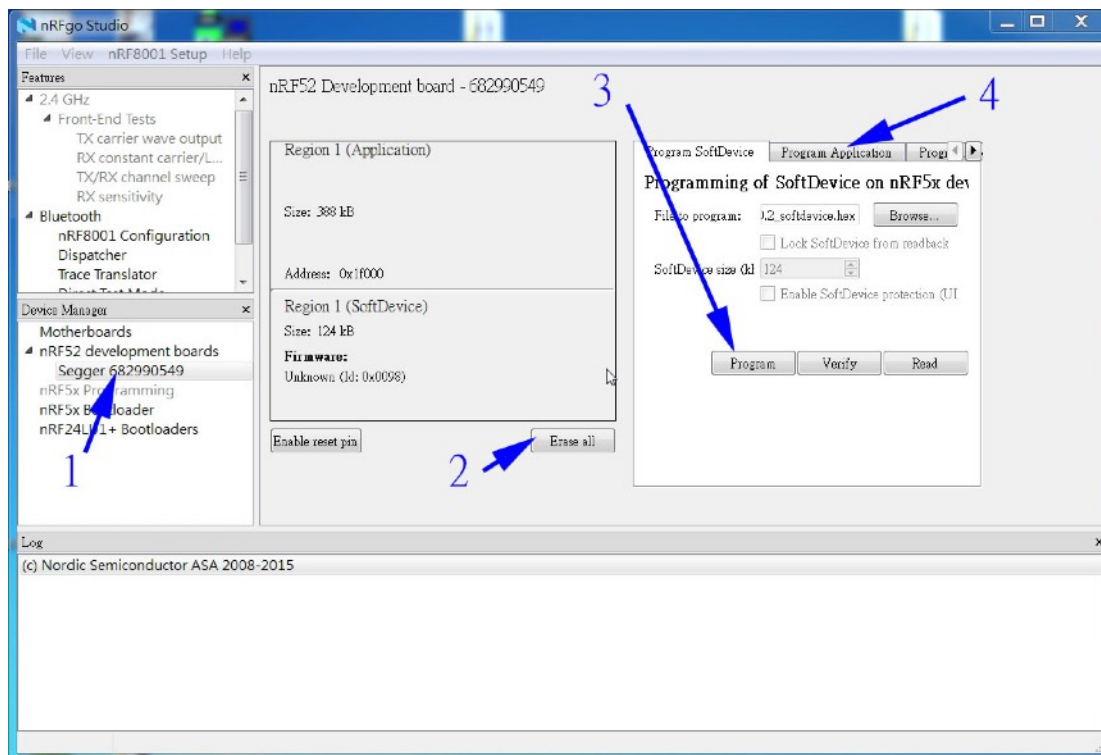
After connecting nRF52DK and an nRF52 sensor/evaluation board, you can see You can J-Link device and COM port on the PC control panel Device Manager.



## Starting Nordic nRFgo

Open nRFgo Studio.

1. Click Segger xxxx
2. Erase all if needed
3. Browse the SoftDevice and program.
4. Browse the application hex file and program



## Over-The-Air DFU

The nRF52 SoC is supported by an Over-The-Air Device Firmware Upgrade (OTA DFU) feature. This allows for in the field updates of application software and SoftDevice.

## Control Skyworks Power Amplifier

BWG832X and BWG832XE use SKYWORKS SKY66112-11 power amplifier. The connection diagram with nRF52832 control signal pins is below.

A firmware configuration example to control Skyworks SKY66112 power amplifier is below. This firmware file, SKY66112\_PAconfig.txt can be downloaded from <http://www.fanstel.com/download-document/>.

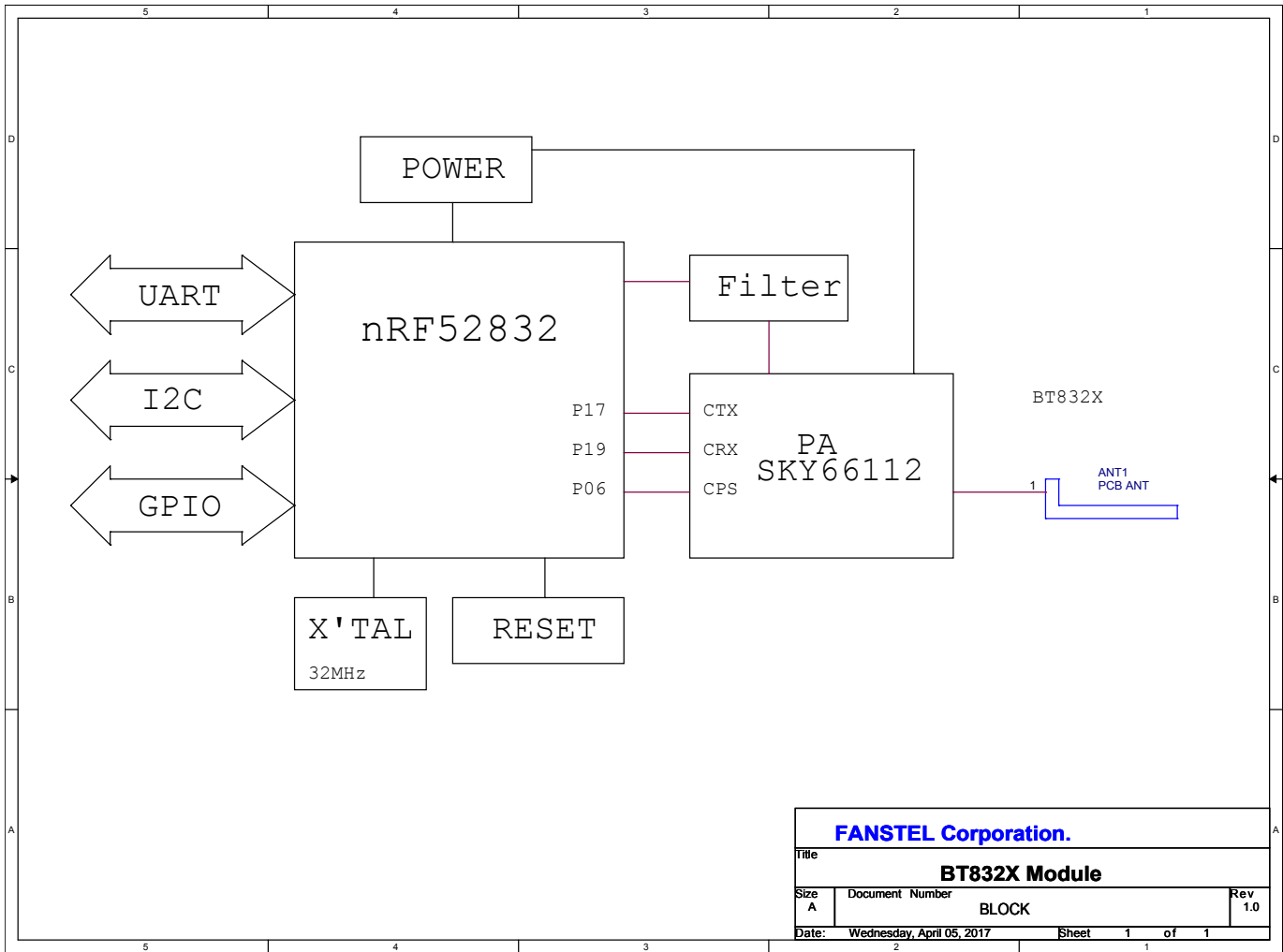
```
//main.c
```

```
//config the PA/LNA
```

```
#ifdef APP_PA_LAN
```

```
#define APP_PA_PIN          17
```

```
#define APP_LNA_PIN         19
```



```
#define APP_CPS_PIN 6
```

```
#define APP_AMP_PPI_CH_ID_SET 0
```

```
#define APP_AMP_PPI_CH_ID_CLR 1
```

```
#define APP_AMP_GPIOTE_CH_ID 0
```

```
static void pa_Ina_setup(void)
```

```
{
```

```
uint32_t err_code;
```

```
nrf_gpio_cfg_output(APP_CPS_PIN);
```

```
    nrf_gpio_pin_clear(APP_CPS_PIN); //enable
```

```
    nrf_gpio_cfg_output(APP_PA_PIN);
```

```
nrf_gpio_pin_clear(APP_PA_PIN); //  
nrf_gpio_cfg_output(APP_LNA_PIN);  
nrf_gpio_pin_clear(APP_LNA_PIN); //
```

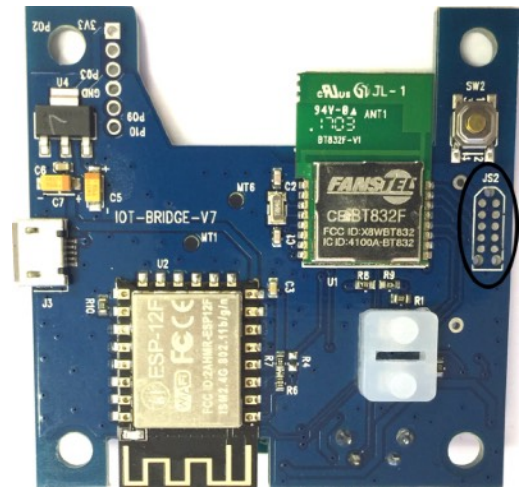
```
static ble_opt_t pa_lna_opts = {  
    .common_opt = {  
        .pa_lna = {  
  
            .pa_cfg = {  
                .enable = 1,  
                .active_high = 1,  
                .gpio_pin = APP_PA_PIN  
            },  
  
            .lna_cfg = {  
                .enable = 1,  
                .active_high = 1,  
                .gpio_pin = APP_LNA_PIN  
            },  
  
            .ppi_ch_id_set = APP_AMP_PPI_CH_ID_SET,  
            .ppi_ch_id_clr = APP_AMP_PPI_CH_ID_CLR,  
            .gpiote_ch_id = APP_AMP_GPIOTE_CH_ID  
        }  
    }  
};  
NRF_GPIO->DIRSET |= (1 << APP_PA_PIN) | (1 << APP_LNA_PIN);  
err_code = sd_ble_opt_set(BLE_COMMON_OPT_PA_LNA, &pa_lna_opts);  
APP_ERROR_CHECK(err_code);
```

```
}  
#endif
```

## Programming Bluetooth Module in Production Gateway

In production gateway (non-Development Kit), JS3 is not installed. The 10-pin flat cable included in DK can not be used to connect the **Debug out** of nRF52DK. A few options on programming Bluetooth module:

- Use a TAG connect cable **TC2050-IDC-NL-050-ALL** to connect **JS2** of gateway to nRF52DK **Debug out** to load bootloader and other firmware.
- If boot loader is loaded, SoftDevice and application firmware can be loaded using OTA.
- For high volume production, Bluetooth module codes can be preloaded at factory.







## 6. Miscellaneous

### **FCC LABEL**

The Original Equipment Manufacturer (OEM) must ensure that the OEM modular transmitter must be labeled with its own FCC ID number. This includes a clearly visible label on the outside of the final product enclosure that displays the contents shown below. If the FCC ID is not visible when the equipment is installed inside another device, then the outside of the device into which the equipment is installed must also display a label referring to the enclosed equipment

The end product with this module may subject to perform FCC part 15 unintentional emission test requirement and be properly authorized.

This device is intended for OEM integrator only.

## Revision History

- June 2018, Ver. 0.50: Initial draft copy release
- July 2018, Ver 0.60:, add FCC and IC certification numbers

## Contact Us

### United States:

Fanstel Corp.

7466 E. Monte Cristo Ave. Scottsdale AZ 85260

Tel. 1 480-948-4928

Fax. 1-480-948-5459

Email: [info@fanstel.com](mailto:info@fanstel.com)

Website: [www.fanstel.com](http://www.fanstel.com)

### Taiwan:

Fanstel Corp.

10F-10, 79 Xintai Wu Road

Xizhu, New Taipei City, Taiwan 22101

泛世公司

臺灣省新北市汐止區新臺五路79號10樓之10, 22101

Tel. 886-2-2698-9328

Fax. 886-2-2698-4813

Email: [info@fanstel.com](mailto:info@fanstel.com)

Website: [www.fanstel.com](http://www.fanstel.com)

### China:

Fanstel Technologies Corp.

11 Jiale Street

Ping-Dih, Long-Gang, Shen Zhen, GD 518117

泛世康科技(深圳)有限公司

廣東省深圳市龍崗區坪地鎮佳樂街11號

Tel. 86-755-8409-0928

Fax. 86-755-8409-0973

QQ. 3076221086

Email: [info@fanstel.com](mailto:info@fanstel.com)

Website: [www.fanstel.com](http://www.fanstel.com)